TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843

THEMIS

12

TEXAS A&M UNIVERSITY
PROJECT THEMIS

Technical Report No. 50

STATISTICAL PERT: DECOMPOSING A PROJECT NETWORK

by

R. L. Sielken Jr. and N. E. Fisher

Texas A&M Research Foundation
Office of Naval Research
Contract N00014-68-A-0140
Project NR047-700

ATTACHMENT I

STATISTICAL PERT: DECOMPOSING A PROJECT NETWORK

by

R. L. Sielken Jr. and N. E. Fisher

THEMIS OPTIMIZATION RESEARCH PROGRAM
Technical Report No. 50

INSTITUTE OF STATISTICS
Texas A&M University
November 1975

Research conducted through the
Texas A&M Research Foundation
and sponsored by the
Office of Naval Research
Contract N00014-68-A-0140
Project NR047-700

ATTACHMENT II

TABLE OF CONTENTS

ABSTRACT

Statistical PERT is a new procedure for obtaining information about the distribution of a project's completion time when the project is comprised of a large number of activities and the time required to complete an individual activity once it can be begun is a random variable. The project is represented as an acyclic network whose arcs correspond to the project activities. This network is simplified by replacing various activity configurations by single equivalent activities and then decomposed into several subnetworks. The distribution and moments of each subnetwork's completion time are bounded and approximated on the basis of two points from each activity's completion time distribution by using some mathematical programming techniques and a new result in the theory of networks. The project's completion time distribution is then approximated by combining the approximate subnetwork distributions.

This report documents two computer programs. The first program BREAKUP decomposes a project network into several subnetworks which are connected in either series or parallel in the project network. The second program LOOP checks a given project network for loops (cycles) since any loop would contradict the assumed acyclic structure of the project network.

BREAKUP

General Description:

This program "breaks up" a network into a set of subnetworks which can be linked together either in series or parallel to yield the given network. This breakup is complete in the sense that none of the subnetworks in the set can be further broken up.

The basic breakup procedure involves two main subroutines, BUNDLE and CUT. BUNDLE partitions the activities in a given network or subnetwork into parallel subnetworks connecting the network's source and sink. CUT identifies the cut nodes in a given network or subnetwork and then identifies the sets of activities between each of the consecutive cut nodes. This series of activity sets represents a breakup of the given network or subnetwork into subnetworks in series. The complete breakup of the given network is the following sequential procedure:

(1) Use BUNDLE to identify the parallel subnetworks connecting the source and sink of the network..

(2) Use CUT separately on each parallel subnetwork identified in the previous step - (1) or (3) -, and breakup the parallel subnetwork into subnetworks in series. If no such breakup is possible for a parallel subnetwork, that subnetwork is not considered again. If no new series subnetwork is identified in this entire step, stop. Otherwise go to (3).

(3) Use BUNDLE separately on each series subnetwork identified in (2). If BUNDLE cannot breakup a series subnetwork, that subnetwork is not considered again. If no new parallel subnetwork is identified in this entire step,

A schematic example of this procedure accompanies the sample problem.

Specific Input Instructions:

Card 1. Col. 1-3: The number of arcs in the network, Format (I3).

Col. 4-6: The number of the node which is the source node, Format (I3).

Col. 7-9: The number of the node which is the sink node, Format (I3).

Col. 10-12: The largest node number in the network, Format (I3).

For each activity one card with:

Col. 1-3: The activity's number, Format (I3).

Col. 4-6: The activity's origin node number, Format (I3).

Col. 7-9: The activity's terminal node number, Format (I3).

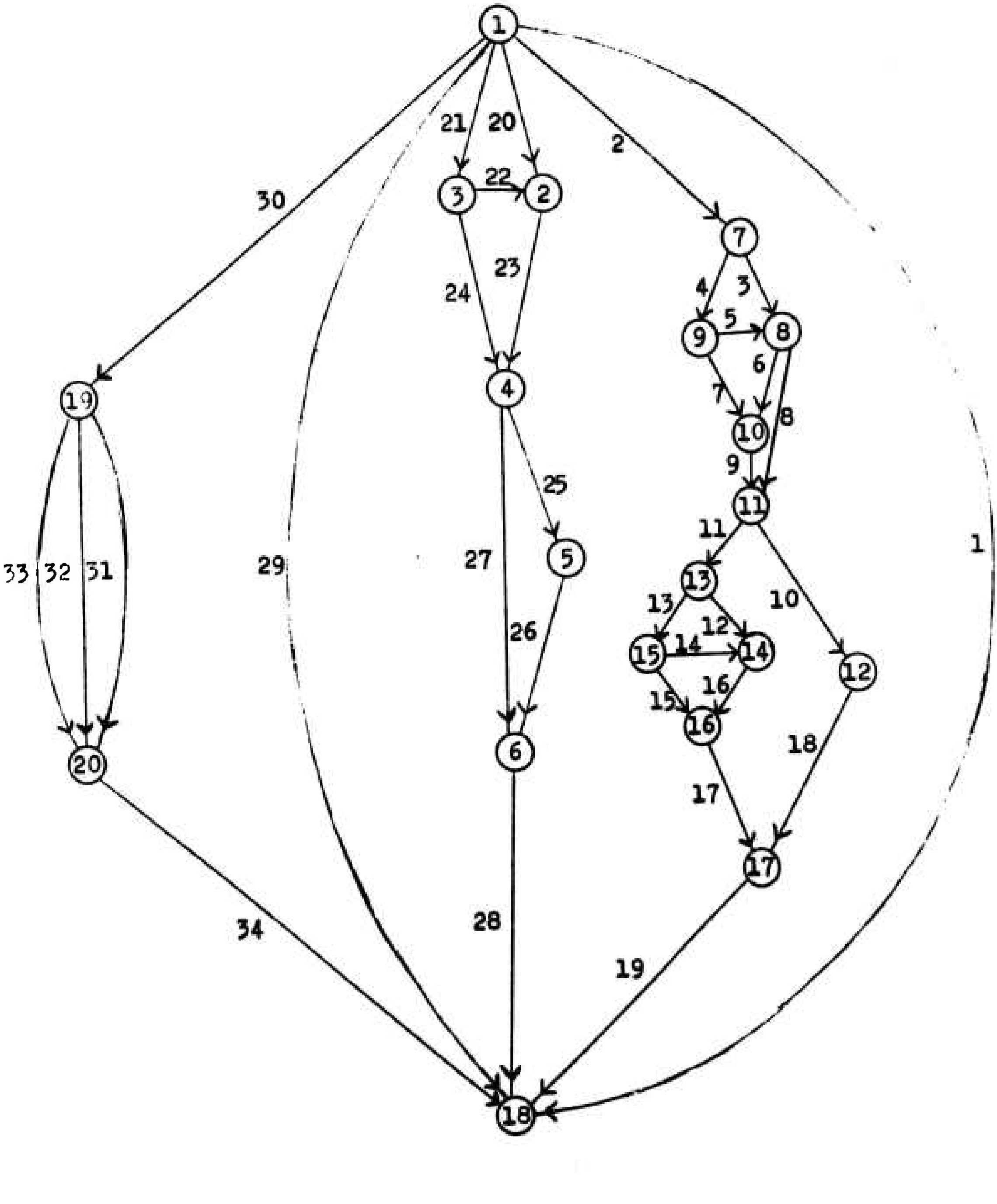
The activities and nodes may be numbered in any way and may be read in in any order.

Dimension Restrictions:

Currently this program will accomodate a network that has a maximum of 300 nodes and a maximum of 300 arcs. It can store a maximum of 100 subnetworks with each subnetwork having a maximum of 300 arcs.

This program is written in FORTRAN G.

SAMPLE NETWORK



SAMPLE INPUT

34	1	18	20
2	1	7	
15	15	16	
12	13	14	
27	4	6	
3	7	8	
28	6	18	
29	1	13	
4	7	9	
5	9	8	
13	13	15	
6	8	10	
16	14	16	
17	16	17	
14	15	14	
7	9	10	
18	12	17	
30	1	19	
19	17	18	
8	8	11	
20	1	2	
31	19	20	
21	1	3	
9	10	11	
10	11	12	
22	3	2	
23	2	4	
32	19	20	
33	19	20	
34	20	18	
24	3	4	
25	4	5	
26	5	6	
1	1	18	
11	11	13	

INPUT STAGE

THE INITIAL NETWORK HAS 34 ARCS

THE SOURCE IS NODE NUMBER 1

THE SINK IS NODE NUMBER 18

THE LARGEST NODE IS NODE NUMBER 20

THE INITIAL NETWORK AS READ IN IS:

ARC NUMBER	ORIGIN NODE	TERMINAL NODE
2	1	7
15	15	16
12	13	14
27	4	6
3	7	8
28	6	18
29	1	18
4	7	9
5	9	8
13	13	15
6	8	10
16	14	16
17	16	17
14	15	14
7	9	10
18	12	17
30	1	19
19	17	18
8	8	11
20	1	2
31	19	20
21	1	3
9	10	11
10	11	12
22	3	2
23	2	4
32	19	20
33	19	20
34	20	18
24	3	4
25	4	5
26	5	6
1	1	18
11	11	13

SUBNETWORK 1 IS COMPOSED OF SUBNETWORKS:

2, 3, 4, 5, 6,

IN PARALLEL

STAGE 1 BREAKUP

SUBNETWORK 2 IS COMPOSED OF SUBNETWORKS:

7, 8, 9, 10,

IN SERIES

SUBNETWORK 3 IS COMPOSED OF SUBNETWORKS:

11, 12, 13,

IN SERIES

SUBNETWORK 4 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 18

ARC	S(ARC)	T(ARC)
29	1	18

SUBNETWORK 5 IS COMPOSED OF SUBNETWORKS:

14, 15, 16,

IN SERIES

SUBNETWORK 6 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 18

ARC	S(ARC)	T(ARC)
1	1	18

STAGE 2 BREAKUP

SUBNETWORK 7 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 7

ARC	S(ARC)	T(ARC)
2	1	7

SUBNETWORK 8 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 7

SINK NODE = 11

ARC	S(ARC)	T(ARC)
3	7	8
4	7	9
6	8	10
8	8	11
5	9	8
7	9	10
9	10	11

SUBNETWORK 9 IS COMPOSED OF SUBNETWORKS:

17, 18,

IN PARALLEL

SUBNETWORK 10 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 17

SINK NODE = 18

ARC	S(ARC)	T(ARC)
19	17	18

SUBNETWORK 11 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 4

ARC	S(ARC)	T(ARC)
20	1	2
21	1	3
23	2	4
22	3	2
24	3	4

SUBNETWORK 12 IS COMPOSED OF SUBNETWORKS:

19, 20,

IN PARALLEL

SUBNETWORK 13 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 6

SINK NODE = 18

ARC	S(ARC)	T(ARC)
28	6	18

SUBNETWORK 14 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 19

ARC	S(ARC)	T(ARC)
30	1	19

SUBNETWORK 15 IS COMPOSED OF SUBNETWORKS:

21, 22, 23,

IN PARALLEL

SUBNETWORK 16 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 20

SINK NODE = 18

ARC	S(ARC)	T(ARC)
34	20	18

STAGE 3 BREAKUP

SUBNETWORK 17 IS COMPOSED OF SUBNETWORKS:

24, 25,

IN SERIES

SUBNETWORK 18 IS COMPOSED OF SUBNETWORKS:

26, 27, 28,

IN SERIES

SUBNETWORK 19 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 4

SINK NODE = 6

ARC	S(ARC)	T(ARC)
27	4	6

SUBNETWORK 20 IS COMPOSED OF SUBNETWORKS:

29, 30,

IN SERIES

SUBNETWORK 21 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 19

SINK NODE = 20

ARC	S(ARC)	T(ARC)
31	19	20

SUBNETWORK 22 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 19

SINK NODE = 20

ARC	S(ARC)	T(ARC)
32	19	20

SUBNETWORK 23 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 19

SINK NODE = 20

ARC	S(ARC)	T(ARC)
33	19	20

STAGE 4 BREAKUP

SUBNETWORK 24 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 11

SINK NODE = 12

ARC	S(ARC)	T(ARC)
10	11	12

SUBNETWORK 25 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 12

SINK NODE = 17

ARC	S(ARC)	T(ARC)
18	12	17

SUBNETWORK 26 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 11

SINK NODE = 13

ARC	S(ARC)	T(ARC)
11	11	13

SUBNETWORK 27 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 13

SINK NODE = 16

ARC	S(ARC)	T(ARC)
12	13	14
13	13	15
16	14	16
15	15	16
14	15	14

SUBNETWORK 28 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

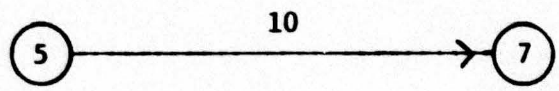
SOURCE NODE = 16

```
     SINK NODE    =  17

     ARC  S(ARC)  T(ARC)
      17    16      17


SUBNETWORK  29 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

     SOURCE NODE =   4

     SINK NODE    =   5

     ARC  S(ARC)  T(ARC)
      25     4       5


SUBNETWORK  30 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

     SOURCE NODE =   5

     SINK NODE    =   6

     ARC  S(ARC)  T(ARC)
      26     5       6
```

SAMPLE PROBLEM: SCHEMATIC REPRESENTATION

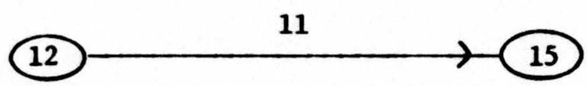
BREAKUP

Arcs are designated as follows:

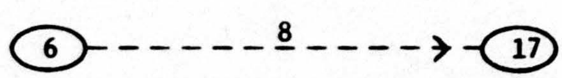


The arc number is written above or to the left of the arc activity line. The arrowhead indicates the direction of the activity. The circled numbers are node numbers. In this case the arc depicted is arc number 10, flowing from left to right with beginning node 5 and terminating node 7.

Subnetworks are designated as follows:



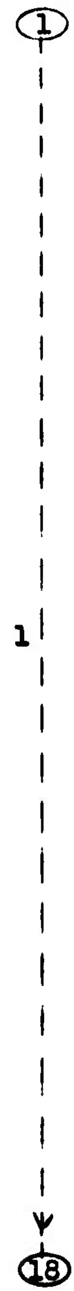
The subnetwork number is written above or to the left of the network activity direction line. The arrowhead indicates the direction of the subnetwork activity. The numbers enclosed in ovals are source and sink nodes. In this case the subnetwork depicted is subnetwork 11, flowing from left to right with source node 12 and sink node 15. A solid activity direction line indicates the subnetwork is a minimum network. A broken line indicates the subnetwork must be considered by at least one more subroutine of the BREAKUP program.



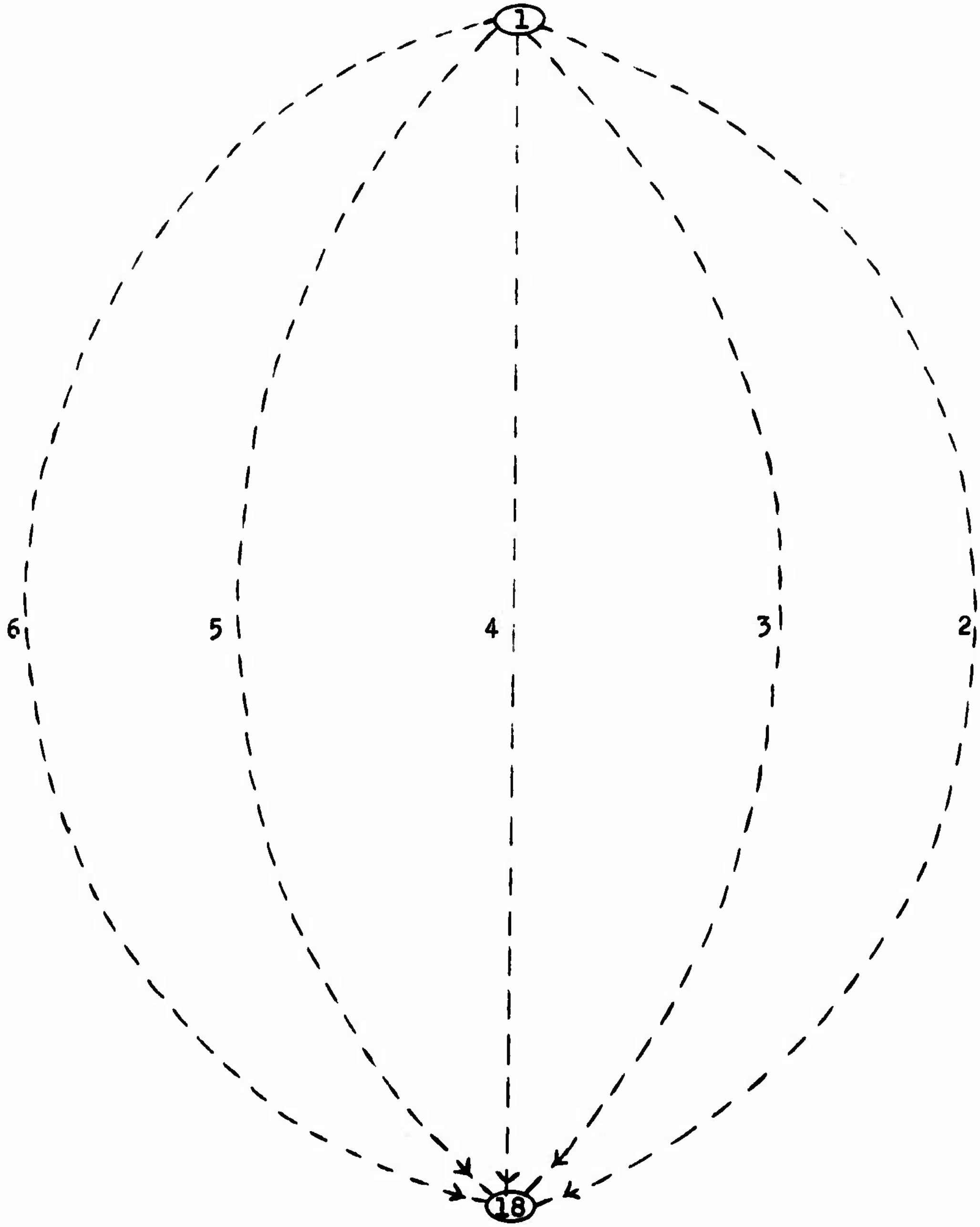
The above figure indicates subnetwork 8, flowing from left to right with source node 6 and sink node 17, is not yet a minimum network.

In the BREAKUP diagrams, the decomposition of the initial network is finished when all subnetworks are minimum networks, e.g. all lines are solid.

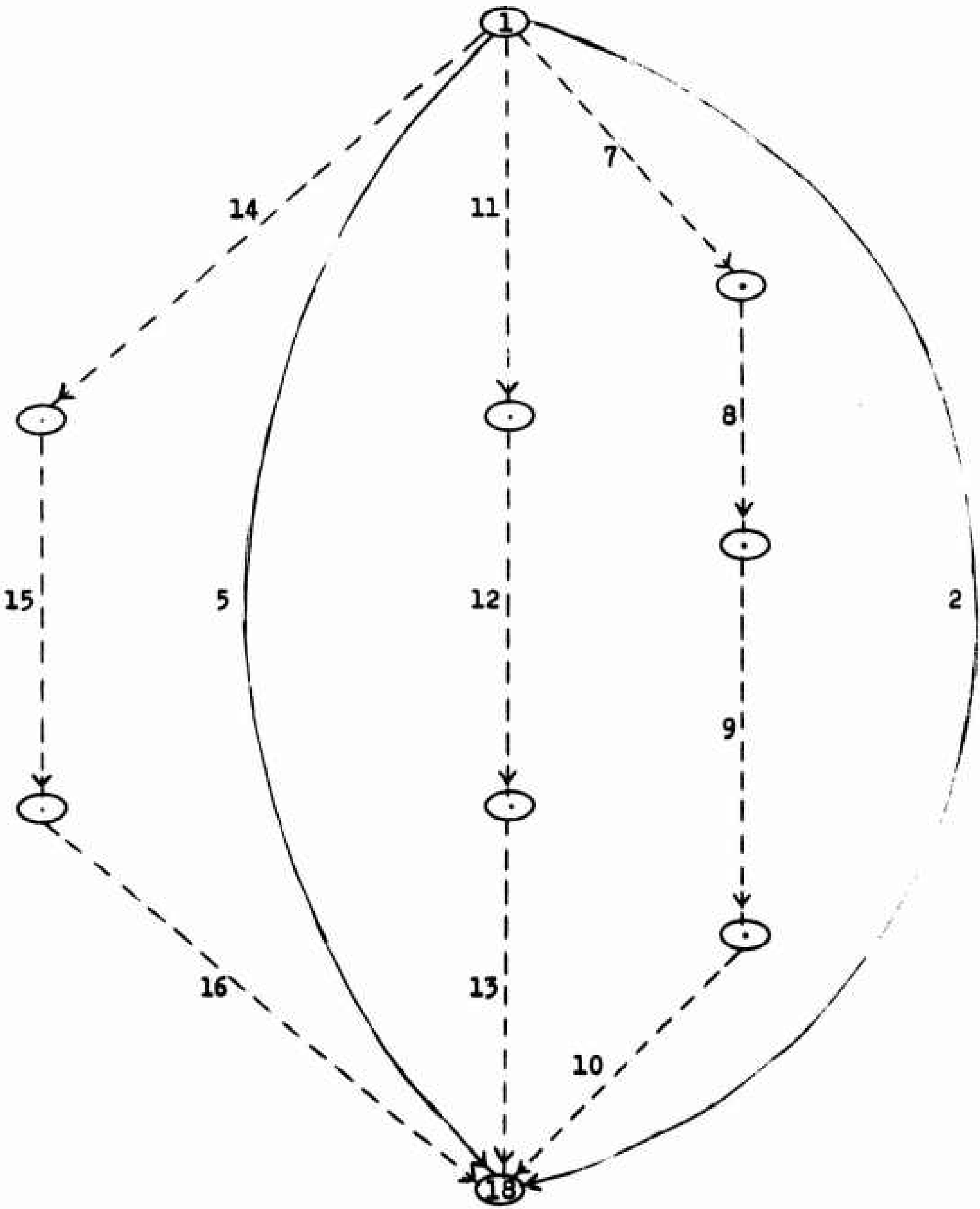
INITIAL NETWORK



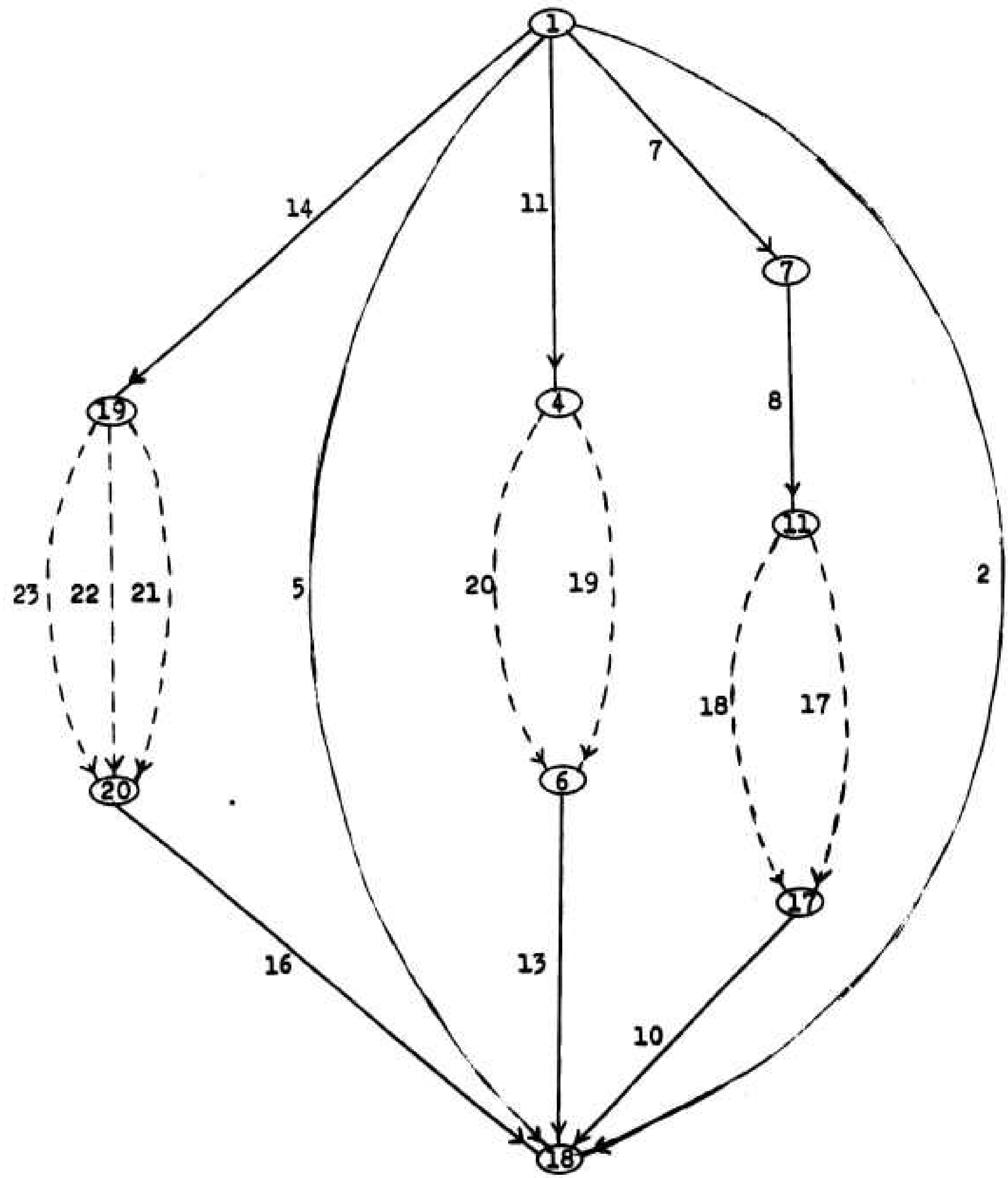
INPUT STAGE



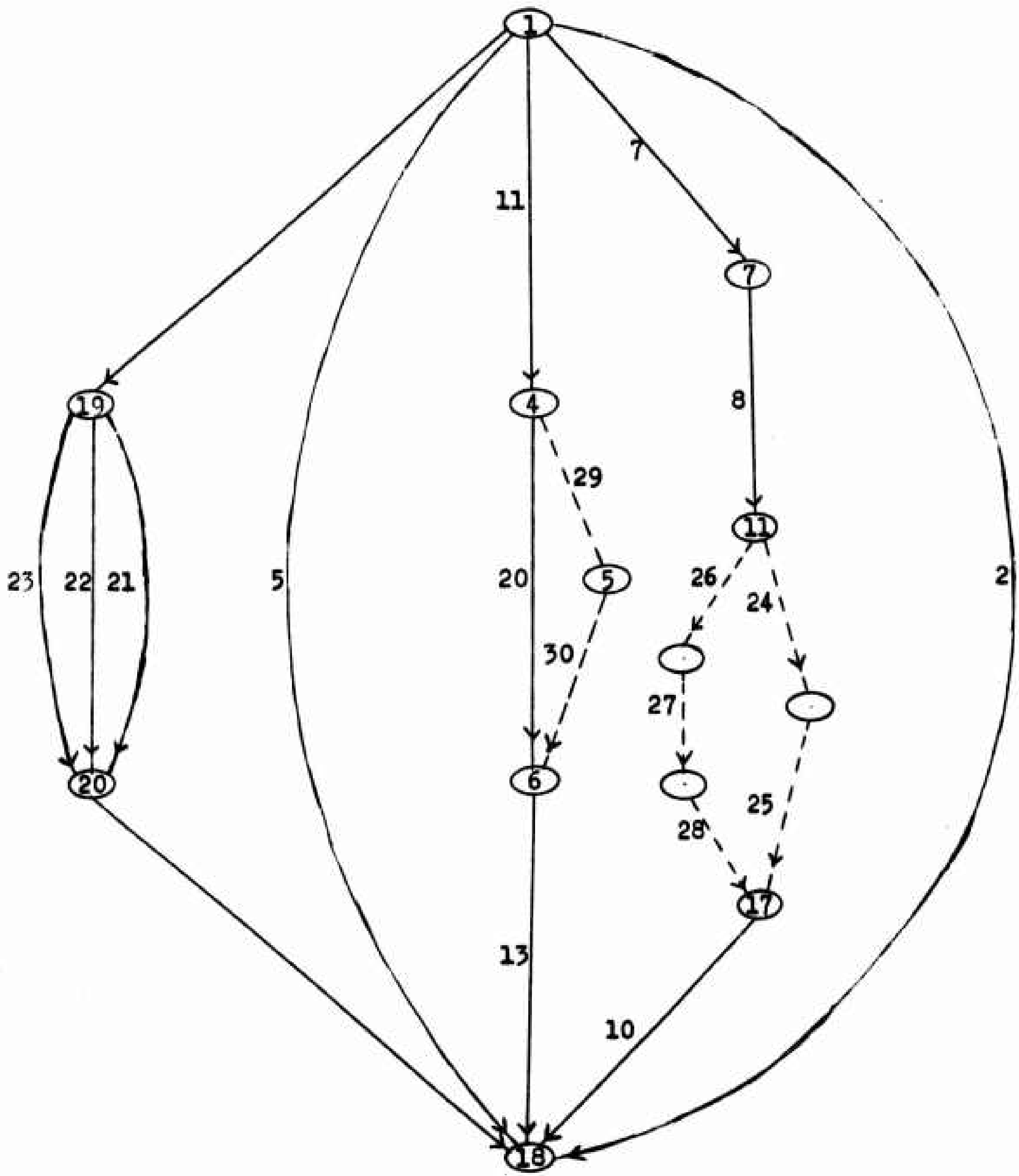
STAGE 1



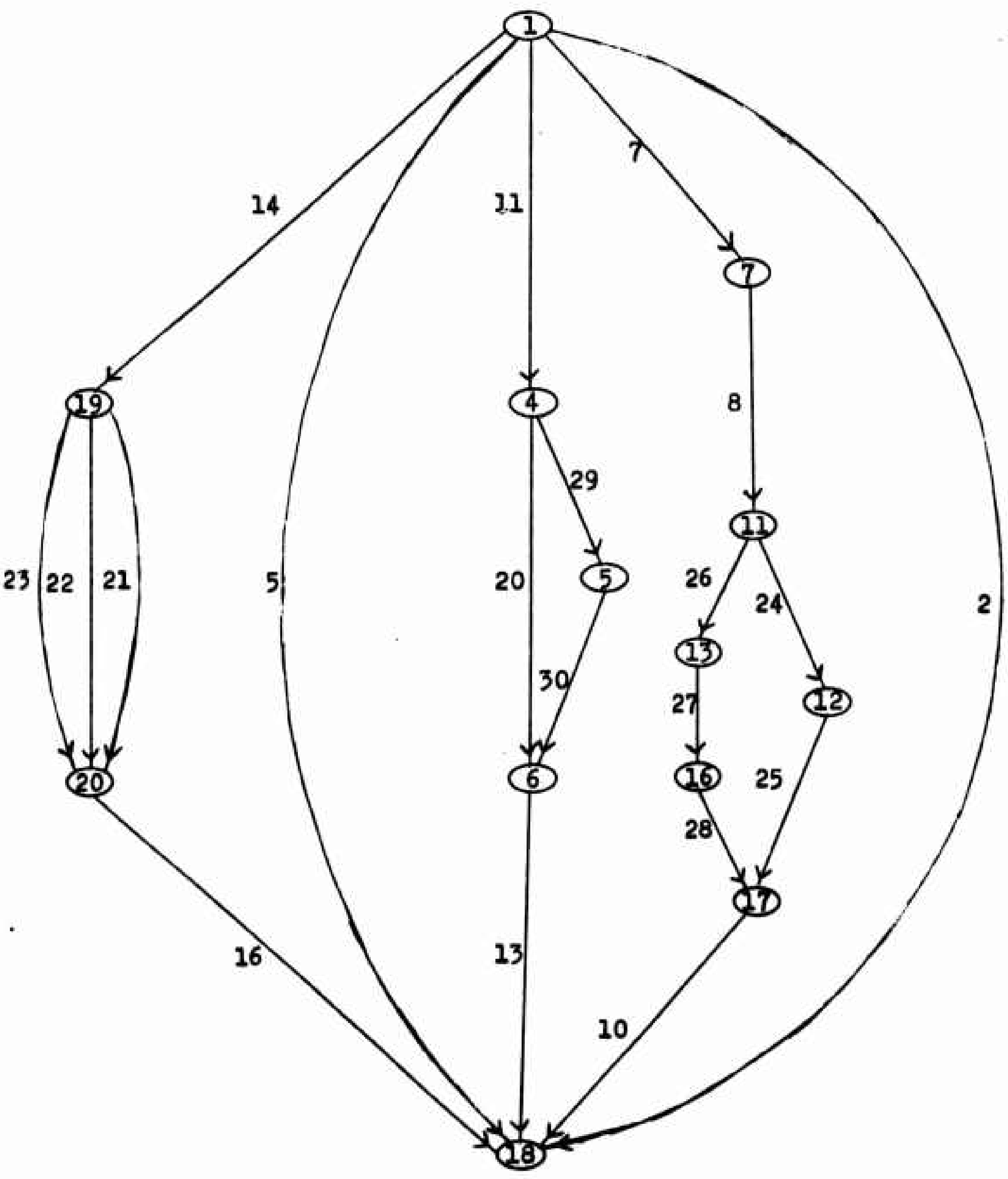
STAGE 2



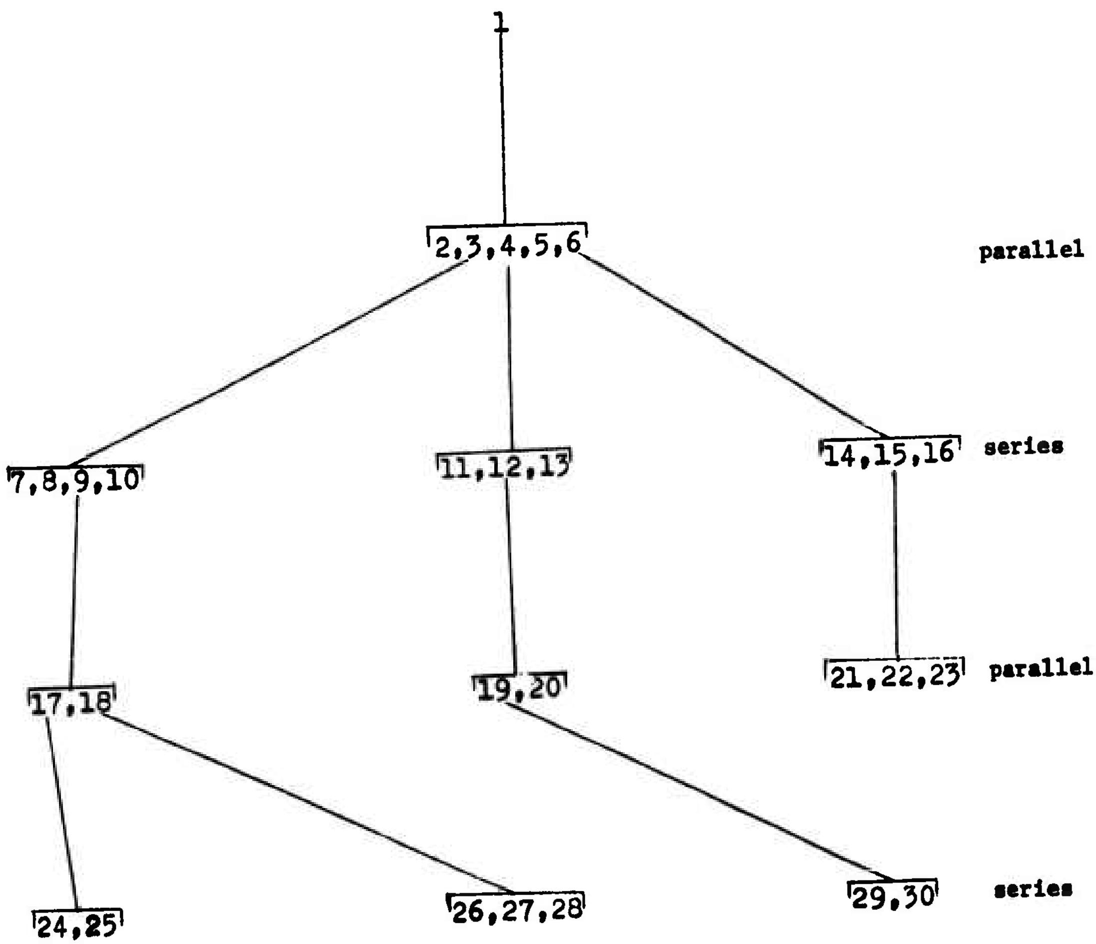
STAGE 3



STAGE 4


1
14
11
7
7
19
4
8
29
11
23
22
21
5
20
5
26
24
2
13
30
12
27
20
6
16
25
28
17
16
13
10
18

BREAKUP FLOW



```
C       PROGRAM BREAKUP
C
C       BREAKUP IDENTIFIES PARALLEL SUBNETWORKS AND SERIES SUBNETWORKS
C
C       MORE THAN ONE NETWORK MAY BE DECOMPOSED DURING A RUN.
C       SEPARATE THE DATA FOR EACH SUBNETWORK BY A BLANK CARD
C
C
C       THE FOLLOWING IS AN ALPHABETICAL LISTING OF THE VARIABLES AND
C       ARRAYS THAT ARE USED IN THIS MAIN PROGRAM AND ITS SUBROUTINES
C
C
C       ARCS = THE NUMBER OF ARCS IN THE SUBNETWORK
C       BDNUM(I) = THE BUNDLE NUMBER TO WHICH NODE I IS ASSIGNED
C       CHECK = ARRAY USED TO STORE ARCS HAVING THE SOURCE AND THE
C               SINK AS THEIR ONLY NODES
C       CTNSUB = THE TEMPORARY NUMBER OF SUBNETWORKS FOUND IN THE
C                PREVIOUS STEP
C       LNODEN = THE LARGEST NODE NUMBER BEING READ IN
C       MAXND = THE LARGEST NODE NUMBER THAT HAS ALREADY BEEN ASSIGNED
C            AT LEAST TEMPORARILY TO A BUNDLE
C       NARCSS(I) = THE NUMBER OF ARCS IN SUBNETWORK I
C       NSUB = THE TOTAL OF SUBNETWORKS THUS FAR
C       NTARC = THE NUMBER OF ARCS IN THIS SUBNETWORK
C       NUMBD = THE NUMBER OF BUNDLES CREATED
C       S(I) = THE STARTING NODE FOR ARC I
C       SINK   = THE NODE NUMBER CORRESPONDING TO THE SINK
C       SINKS(I) = THE SINK IN SUBNETWORK I
C       SOURC(I) = SOURCE NODE IN SUBNETWORK I
C       SOURCE = THE NODE NUMBER CORRESPONDING TO THE SOURCE
C       STEP = STAGE NUMBER
C       SUBNET(I,J) = THE ITH ARC IN THE JTH SUBNETWORK
C       SUMARC = THE CURRENT NUMBER OF ARCS IN SUBNETWORK NSUB
C       T(I) = THE TERMINATING NODE FOR ARC I
C       TARC = THE SUBNETWORK WITHOUT THE ARCS INVCLVING NODE K
C       TLNDEN = TEMPORARY LARGEST NODE NUMBER
C       TNSUB = THE NUMBER OF SUBNETWORKS CREATED IN THE CURRENT STAGE
C       TSNSUB = THE TEMPORARY SUBNETWORK BEING USED IN STAGE
C                SUBROUTINE
C       TSUBN = THE NUMBER OF THE SUBNETWORK CURRENTLY BEING CONSIDERED
C       TYPESN = THE TYPE OF SUBNETWORK BEING CONSIDERED:
C            1 = BUNDLE SUBNETWORK  2 = CUT SUBNETWORK
C
C       A MINIMUM SUBNETWORK IS ONE THAT IS NCT COMPOSED OF SMALLER
C       SUBNETWORKS
C
      IMPLICIT INTEGER*2 (A-Z)
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)
      DIMENSION SINKS(100),NARCSS(100)
C
C       THIS PROGRAM WILL ACCCMODATE A NETWORK THAT HAS A MAXIMUM OF
C       300 NODES, A MAXIMUM OF 300 ARCS AND CAN STORE A MAXIMUM OF
C       100 SUBNETWORKS IN THE BREAKUP PROCESS
C
C       NA = THE NUMBER OF ARCS IN THE NETWORK TO BE CONSIDERED
C       NS = THE TOTAL NUMBER OF POSSIBLE SUBNETWORKS IN THE BREAKUP
C            PROCESS
C       IN THE CASE WHERE NO GOOD ESTIMATE CAN BE MADE OF NS, USE NA
```

```
C        FOR ALL DIMENSIONS
C        THE ARRAY DIMENSIONS ARE:
C        S(NA),T(NA),SUBNET(NA,NS),SOURC(NS),SINKS(NS),NARCSS(NS)
C
C        IN SUBROUTINE BUNDLE THE ARRAY DIMENSIONS ARE:
C        BDNUM(NS),CHECK(NS)
C
C        IN SUBROUTINE CUT THE ARRAY DIMENSIONS ARE:
C        ORIGIN(NS),POST(NS),TARC(NS),RCUT(NA)
C
C        READ THE NETWORK IN
C
6000  CALL NETIN (LNODEN)
      STEP=1
      TYPESN=1
      TSUBN=1
      CALL BUNDLE (LNODEN,TSUBN)
C
C        PRINT OUT THIS STAGE OF THE BREAKUP
C
      CALL STAGE (TYPESN,TSUBN)
      CTNSUB=TNSUB
      GO TO 70
C
C        FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS
C
10    CTNSUB=0
      TSUBN=NSUB-LOOP
      TYPESN=1
      DO 20 I=1,LOOP
C
C        FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED
C
      TSUBN=TSUBN+1
C
C        FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN
C
      CALL NODER (TLNDEN,TSUBN)
C
C        FIND THE BUNDLE SUBNETWORKS
C
      CALL BUNDLE (TLNDEN,TSUBN)
C
C        PRINT OUT THIS STAGE OF THE BREAKUP
C
      CALL STAGE (TYPESN,TSUBN)
C
C        IF THERE IS ONLY ONE BUNDLE FOUND IN SUBNETWORK TSUBN, WE ARE
C        FINISHED.  PRINT OUT ITS COMPONENT ARCS
C
      IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)
C
C        COUNT THE NEW NUMBER OF SUBNETWORKS CREATED
C
20    CTNSUB=TNSUB+CTNSUB
C
C        IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED
C
      IF (CTNSUB.EQ.0) GO TO 90
      GO TO 70
```

```
80     LOOP=CTNSUB
       CTNSUB=0
       TSUBN=NSUB-LOOP
C
C         FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS
C
       TYPESN=2
       DO 30 I=1,LOOP
C
C         FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED
C
       TSUBN=TSUBN+1
       IF (TSUBN.EQ.1) GO TO 55
C
C         FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN
C
       CALL NODER (TLNDEN,TSUBN)
C
C         FIND THE CUT SUBNETWORKS
C
55     IF (TSUBN.EQ.1) TLNDEN=LNODEN
       CALL CUT (TSUBN,TLNDEN)
C
C         PRINT OUT THIS STAGE OF THE BREAKUP
C
       CALL STAGE (TYPESN,TSUBN)
C
C         IF THERE ARE NO CUTS FOUND IN SUBNETWORK TSUBN, WE ARE FINISHED
C         PRINT OUT ITS COMPONENT ARCS
C
       IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)
C
C         COUNT THE NEW NUMBER OF SUBNETWORKS CREATED
C
30     CTNSUB=TNSUB+CTNSUB
C
C         IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED
C
       IF (CTNSUB.EQ.0) GO TO 90
       GO TO 70
85     LOOP=CTNSUB
       GO TO 10
70     WRITE (6,900) STEP
900    FORMAT (1H1,5X,'STAGE',I3,' BREAKUP')
       STEP=STEP+1
C
C         LET'S GO BACK TO THE APPROPRIATE LOOP FOR THE NEXT STAGE
C
       GO TO (80,85),TYPESN
90     CONTINUE
       READ (5,100,END=666)
100    FORMAT (I3)
       GO TO 6000
666    CONTINUE
       WRITE (6,9000)
9000   FORMAT (1H1)
       RETURN
       END

       SUBROUTINE NETIN (LNODEN)
```

```
      IMPLICIT INTEGER*2 (A-Z)
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)
      DIMENSION SINKS(100),NARCSS(100)
C
C        ZEROIZE SOURC ARRAY
C
      DO 20 I=1,50
20    SOURC(I)=0
C
C        READ IN THE INITIAL NETWORK LIMITS
C
      READ (5,100) ARCS,SOURCE,SINK,LNODEN
100   FORMAT (4I3)
      WRITE (6,200)
200   FORMAT (1H1,5X,'INPUT STAGE')
      WRITE (6,210) ARCS,SOURCE,SINK,LNODEN
210   FORMAT (1H0,/6X,'THE INITIAL NETWORK HAS',12X,I4,' ARCS',//6X,
     *'THE SOURCE IS NODE NUMBER',11X,I3,//6X,'THE SINK IS NODE NUMBER',
     *13X,I3,//6X,'THE LARGEST NODE IS NODE NUMBER',5X,I3)
      WRITE (6,220)
220   FORMAT (1H0,5X,'THE INITIAL NETWORK AS READ IN IS:',///6X,
     *'ARC NUMBER',5X,'ORIGIN NODE',5X,'TERMINAL NODE')
C
C        READ IN EACH ARC AND ITS STARTING AND TERMINATING NODES
C        THE ARCS AND NODES MAY BE NUMBERED ANY WAY AND READ IN IN ANY
C        ORDER
C        I = ARC NUMBER
C        S = THE NODE NUMBER FOR THE START OF AN ARC
C        T = THE TERMINAL NODE OF AN ARC
C
      DO 10 J=1,ARCS
      READ (5,100) I,S(I),T(I)
      WRITE (6,240) I,S(I),T(I)
240   FORMAT (1H ,8X,I3,13X,I3,14X,I3)
C
C        CREATE THE FIRST SUBNETWORK
C
10    SUBNET (J,1)=I
      SOURC(1)=SOURCE
      SINKS(1)=SINK
      NARCSS(1)=ARCS
      NSUB=1
      RETURN
      END

      SUBROUTINE NODER (TLNDEN,TSUBN)
C
C        FINDS LARGEST NODE NUMBER IN THE SUBNETWORK TSUBN
C
      IMPLICIT INTEGER*2 (A-Z)
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)
      DIMENSION SINKS(100),NARCSS(100)
      ARCS=NARCSS(TSUBN)
      TLNDEN=0
      DO 20 J=1,ARCS
      A=SUBNET(J,TSUBN)
      M=S(A)
      N=T(A)
```

```
      MAXND=N
      IF (M.GT.N) MAXND=M
      IF (MAXND.GT.TLNDEN) TLNDEN=MAXND
20    CONTINUE
      RETURN
      END

      SUBROUTINE ENDSNT (TSUBN)
C
C        PRINTS SMALLEST BREAKDCWN OF SUBNETWORK TSUBN
C
      IMPLICIT INTEGER*2 (A-Z)
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)
      DIMENSION SINKS(100),NARCSS(100)
      SOURCE=SOURC(TSUBN)
      SINK=SINKS(TSUBN)
      WRITE (6,100) TSUBN
100   FORMAT (1H0,//16X,'SUBNETWORK ',I3,' IS A MINIMUM NETWORK',/16X,
     *'IT IS COMPOSED OF:')
      WRITE (6,200) SOURCE,SINK
200   FORMAT (1H0,19X,'SOURCE NODE = ',I3,//20X,'SINK NODE   = ',I3)
      M=NARCSS(TSUBN)
      WRITE (6,400)
400   FORMAT (1H0,19X,'ARC',2X,'S(ARC)',2X,'T(ARC)')
      DO 10 I=1,M
      N=SUBNET(I,TSUBN)
10    WRITE (6,300) N,S(N),T(N)
300   FORMAT (1H ,19X,I3,3X,I3,5X,I3)
C
C        SET TNSUB=0 SO THAT THE REMAINING NUMBER OF SUBNETWORKS DOESN'T
C        INCLUDE THIS MINIMUM SUBNETWORK
C
      TNSUB=0
      RETURN
      END

      SUBROUTINE STAGE (TYPESN,TSUBN)
C
C        PRINTS OUT THE CURRENT STAGE OF BREAKUP
C
      IMPLICIT INTEGER*2 (A-Z)
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)
      DIMENSION SINKS(100),NARCSS(100)
C
C        TNSUB=THE NUMBER OF NEW SUBNETWORKS RESULTING FROM THE BREAKUP
C        IN THIS STAGE
C
C        TNSUB=1 IMPLIES NO BREAKUP OCCURRED IN THIS STAGE
C
      IF (TNSUB.EQ.1) GO TO 600
      WRITE (6,100) TSUBN
100   FORMAT (1H0,/6X,'SUBNETWORK ',I3,' IS COMPOSED OF SUBNETWORKS: ')
      TSNSUB=NSUB-TNSUB
      M=TSNSUB+1
      WRITE (6,300)(I,I=M,NSUB)
300   FORMAT (1H0,5X,20(I3,','))
      IF (TYPESN.EQ.1) GO TO 60
      WRITE (6,400)
```

```
400     FORMAT (1H0,5X,'IN SERIES')
        GO TO 600
60      WRITE (6,500)
500     FORMAT (1H0,5X,'IN PARALLEL')
600     RETURN
        END

        SUBROUTINE BUNDLE (LNODEN,TSUBN)
C
C          BUNDLE IDENTIFIES PARALLEL SUBNETWORKS CONNECTING DESIGNATED
C          SOURCE AND SINK
C
        IMPLICIT INTEGER*2 (A-Z)
        COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
        DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)
        DIMENSION SINKS(100),NARCSS(100),BDNUM(300),CHECK(300)
C
C          GROUP NODES INTO BUNDLES
C
         SOURCE=SOURC(TSUBN)
         SINK=SINKS(TSUBN)
        NUMBD=1
C
C          ZEROIZE THE BDNUM ARRAY
C
        DO 10 I=1,LNODEN
     10 BDNUM(I)=0
        K=SUBNET(1,TSUBN)
        M=S(K)
        N=T(K)
        BDNUM(M)=1
        BDNUM(N)=1
        MAXND=N
        IF (M.GT.N) MAXND=M
        ARCS=NARCSS(TSUBN)
        IF (ARCS.EQ.1) GO TO 515
        DO 1 K=2,ARCS
        I=SUBNET(K,TSUBN)
        BDNUM(SOURCE)=0
        BDNUM(SINK)=0
        M=S(I)
        N=T(I)
        IF(M.GT.MAXND) MAXND=M
        IF(N.GT.MAXND) MAXND=N
C
C          IF AT LEAST 1 NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A BUNDLE
C          GO TO 2
C
        IF(BDNUM(M).EQ.0) GO TO 2
C
C          IF ONLY THE TERMINAL NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A
C          BUNDLE, GO TO 3
C
        IF(BDNUM(N).EQ.0) GO TO 3
C
C          IF BOTH NODES ON THE ARC HAVE BEEN ASSIGNED TO THE SAME BUNDLE
C          EVERYTHING IS OKAY, GO TRY ANOTHER ARC
C
        IF(BDNUM(N).EQ.BDNUM(M)) GO TO 1
C
```

```
C         IF THE NODES CN THE ARC ARE ASSIGNED TO DIFFERENT BUNDLES,
C         THEN THESE TWO BUNDLES SHOULD BE POOLED
C
      IF(BDNUM(N).LT.BDNUM(M)) GO TO 6
C
C         POOL BUNDLES
C         THE BUNDLE WITH THE LARGER BUNDLE NUMBER IS POOLED INTO THE
C         BUNDLE WITH THE SMALLER BUNDLE NUMBER
C         THE BUNDLE NUMBERS OF ALL BUNDLES ARE ALL ADJUSTED
C
      MAXBD=BDNUM(N)
      MINBD=BDNUM(M)
      GO TO 7
    6 MAXBD=BDNUM(M)
      MINBD=BDNUM(N)
    7 DO 5 J=1,MAXND
      B=BDNUM(J)
      IF (B.EQ.MAXBD) BDNUM(J)=MINBD
      IF (B.GT.MAXBD) BDNUM(J)=BDNUM(J)-1
    5 CONTINUE
      NUMBD=NUMBD-1
      GO TO 1
C
C         IF BOTH NODES ON THE ARC ARE UNASSIGNED, GO TO 4 WHERE A NEW
C         BUNDLE IS CREATED
C
    2 IF(BDNUM(N).EQ.0) GO TO 4
C
C         ASSIGN THE ORIGIN NODE OF THE ARC TO THE BUNDLE CONTAINING THE
C         TERMINAL NODE
C
      BDNUM(M)=BDNUM(N)
      GO TO 1
C
C         ASSIGN THE TERMINAL NODE OF THE ARC TO THE BUNDLE CONTAINING
C         THE ORIGIN NODE OF THE ARC
C
    3 BDNUM(N)=BDNUM(M)
      GO TO 1
C
C         CREATE A NEW BUNDLE
C
    4 NUMBD=NUMBD+1
      BDNUM(M)=NUMBD
      BDNUM(N)=NUMBD
    1 CONTINUE
 515  CONTINUE
      BDNUM(SINK)=0
C
C         IF WE ONLY HAVE 1 BUNDLE FROM THE SUBNETWORK, WE ARE FINISHED
C
      IF (NUMBD.EQ.1) GO TO 219
C
C         ZEROIZE CHECK ARRAY
C
      DO 290 I=1,ARCS
 290  CHECK(I)=0
      L=0
C
C         THE NODES ARE IN BUNDLES.  PUT THE ASSOCIATED ARCS INTO
```

```
C        APPROPRIATE PARALLEL SUBNETWORKS
C
      DO 33 I=1,NUMBD
      SUMARC=0
      NSUB=NSUB+1
      DO 34 K=1,ARCS
      M=SUBNET(K,TSUBN)
      N=S(M)
C
C        SOURCE AND SINK HAVE BUNDLE NUMBER 0
C
      IF (N.EQ.SOURCE) N=T(M)
      IF (BDNUM(N).EQ.I) GO TO 239
      IF (N.EQ.SINK) GO TO 229
      GO TO 34
C
C        SPECIAL CASE:  BUNDLE HAS ONLY 2 NODES: SOURCE, SINK.
C        PUT ALL ARCS THAT ARE PARALLEL SUBNETWORKS BY THEMSELVES INTO
C        THE CHECK ARRAY
C
229   DO 291 J=1,K
      W=CHECK(J)
      IF (W.EQ.0) GO TO 292
      IF (M.EQ.W) GO TO 34
291   CONTINUE
292   CHECK(J)=M
      GO TO 34
C
C        THIS ARC IS IN THE BUNDLE I, HENCE IT IS IN THE ITH NEW
C        SUBNETWORK
C
239   SUMARC=SUMARC+1
      SUBNET(SUMARC,NSUB)=M
34    CONTINUE
C
C        CREATE NEW SUBNETWORKS
C
C
C        IF THIS BUNDLE HAS NO NODES, PUT AN ARC SUBNETWORK INTO
C        SUBNET (1,NSUB)
C
      IF (SUMARC.EQ.0) GO TO 333
343   NARCSS(NSUB)=SUMARC
      SOURC(NSUB)=SOURCE
      SINKS(NSUB)=SINK
      GO TO 33
C
C        STORE THE SUBNETWORKS THAT HAVE ONLY SOURCE AND SINK NODES
C
333   SUMARC=1
      L=L+1
      SUBNET(1,NSUB)=CHECK(L)
      GO TO 343
33    CONTINUE
219   TNSUB=NUMBD
      RETURN
      END

      SUBROUTINE CUT (TSUBN,LNODEN)
C
```

```
C         CUT IDENTIFIES CUT NODES EXCLUDING THE DESIGNATED SOURCE AND
C         SINK
C
C         CUT ALSO IDENTIFIES THE CUT GROUPS; THAT IS, THE SUBNETWORKS
C         WHICH ARE IN SERIES AND CONNECTED BY THE CUT NODES
C
      IMPLICIT INTEGER*2 (A-Z)
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100),TARC(300)
      DIMENSION SINKS(100),NARCSS(100),ORIGIN(300),POST(300),RCUT(100)
C
C         FIND THE CUT NODES
C         NCUT IS THE NUMBER OF CUT NODES FOUND THUS FAR
C
      NCUT=0
      ARCS=NARCSS(TSUBN)
      SOURCE=SOURC(TSUBN)
      SINK=SINKS(TSUBN)
C
C         THE DO LOOP DOWN TO STATEMENT NUMBER 1 DETERMINES THE CUT NODES
C
      DO 1  K=1,LNODEN
C
C         CHECK TO SEE IF NODE K IS ACTUALLY IN THE SUBNETWORK
C
      DO 20 J=1,ARCS
      Z=SUBNET(J,TSUBN)
      IF(S(Z).EQ.K) GO TO 21
      IF (T(Z).EQ.K) GO TO 21
20    CONTINUE
C
C         NODE K IS NOT IN THIS SUBNETWORK
C
      GO TO 1
21    CONTINUE
C
C         NODE K IS IN THIS SUBNETWORK
C
      IF(K.EQ.SOURCE) GO TO 1
      IF(K.EQ.SINK) GO TO 1
      NTARC=0
      DO 2 J=1,ARCS
      Z=SUBNET(J,TSUBN)
      IF (S(Z).EQ.K) GO TO 2
      IF (T(Z).EQ.K) GO TO 2
      NTARC=NTARC+1
      TARC(NTARC)=SUBNET(J,TSUBN)
2     CONTINUE
C
C         TARC IS THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K
C         IF TARC CONTAINS A PATH FROM THE SOURCE TO THE SINK, THEN NODE
C         K IS NOT A CUT NODE
C         OTHERWISE, K IS A CUT NODE
C
      ORIGIN(1)=SOURCE
      NORIG=1
11    CONTINUE
      NPOST=0
C
C         IF THERE ARE NO ARCS IN THE TARC ARRAY, K IS A CUT NODE
```

```
C
      IF (NTARC.EQ.0) GO TO 44
C
C        FIND ALL NODES WHICH CCME AFTER AN ORIGIN; PUT THEM IN POST
      DO 4 I=1,NORIG
      DO 5 J=1,NTARC
      Y=ORIGIN(I)
      Z=TARC(J)
      U=S(Z)
      V=T(Z)
      IF (U.NE.Y) GO TO 5
      IF (V.EQ.SINK) GO TO 1
C
C
C        IF WE'VE REACHED THE SINK, NODE K IS NOT A CUT NODE
C
C
      IF(NPOST.GE.1) GO TO 8
      NPOST=NPOST+1
      POST(NPOST)=V
      GO TO 5
8     DO 9 L=1,NPOST
C
C        IF THIS TERMINAL NODE IS ALREADY A POST, LET'S IGNORE IT
C
      IF (POST(L).EQ.V) GO TO 5
9     CONTINUE
      NPOST=NPOST+1
      POST(NPOST)=V
5     CONTINUE
4     CONTINUE
C
C        IF THERE ARE NOW NO POSTS, NODE K IS A CUT NODE
C
44    IF(NPOST.NE.0) GO TO 13
      NCUT=NCUT+1
      RCUT(NCUT)=K
      GO TO 1
13    NORIG=NPOST
      DO 14 L=1,NORIG
C
C        THESE ARE NOW OUR NEW ORIGINS
C
14    ORIGIN(L)=POST(L)
C
C        CHECK THE NEW ORIGINS FOR THEIR POSTS
C
      GO TO 11
1     CONTINUE
      IF (NCUT.EQ.0) GO TO 32
      NSUB=NSUB+1
      SOURC(NSUB)=SOURCE
33    ORIGIN(1)=SOURCE
C
C        NOW WE NEED TO FIND THE CCMPCNENTS OF THE SERIES SUBNETWORKS
C        THAT ARE SEPARATED BY THE CUT NODES
C
39    SUMARC=0
      NORIG=1
23    NPOST=0
```

```
C
C        ZEROIZE POST ARRAY
C
      DO 55 I=1,LNODEN
55    POST(I)=0
      DO 24 I=1,NORIG
      Y=ORIGIN(I)
      DO 25 J=1,ARCS
      Z=SUBNET(J,TSUBN)
C
C        ALL ARCS BEGINNING AT THIS ORIGIN GO INTO THE NEW SUBNETWORK
C
      IF (S(Z).NE.Y) GO TO 25
      SUMARC=SUMARC+1
      SUBNET (SUMARC,NSUB)=Z
C
C        T(Z) WILL BE A NEW ORIGIN IF IT ISN'T A REPEAT OF A CURRENT
C        ORIGIN
C
C        CHECK TO SEE IF IT IS A REPEAT
C
      DO 30 K=1,NORIG
      X=T(Z)
C
C        IF T(Z) IS A REPEAT OF A CURRENT ORIGIN, LET'S IGNORE IT
C
      IF(X.EQ.ORIGIN(K)) GO TO 25
30    CONTINUE
      NPOST=NPOST+1
      POST(NPOST)=X
C
C        CHECK TO SEE IF ANY POST IS A CUT NODE
C        IF IT IS, REPLACE IT WITH A ZERO
C
      DO 52 W=1,NCUT
      D=RCUT(W)
      IF (X.NE.D) GO TO 52
      POST(NPOST)=0
      NPOST=NPOST-1
C
C        THIS CUT NODE IS THE SINK OF THE SUBNETWORK UNDER
C        CONSIDERATION AND THE SOURCE OF THE NEXT SUBNETWORK TO BE
C        CONSIDERED.
C
      SINKS(NSUB)=D
      SOURC(NSUB+1)=D
C
C        CHECK TO SEE IF ANY POSTS ARE REPEATED IN THE POST ARRAY.  IF
C        THEY ARE, REDUCE THE NUMBER OF POSTS TO WHERE THERE ARE NO
C        REPEATS
C
52    CONTINUE
C
C        IF WE HAVE ONE OR LESS POSTS, THERE ARE NO ADJUSTMENTS OF THE
C        POST ARRAY TO BE MADE; LET'S CONTINUE
C
53    IF (NPOST.LE.1) GO TO 25
      POSTCK=NPOST-1
      DO 7 K=1,POSTCK
      IF (POST(K).EQ.POST(NPOST)) GO TO 61
```

```
7       CONTINUE
        GO TO 25
61      POST(NPOST)=0
        NPOST=NPOST-1
25      CONTINUE
24      CONTINUE
C
C          IF WE HAVE NO POSTS LEFT, WE HAVE FOUND ALL OF THIS SUBNETWORK
C
        IF (NPOST.EQ.0) GO TO 34
        NORIG=NPOST
        DO 28 L=1,NORIG
28      ORIGIN(L)=POST(L)
        GO TO 23
34      NARCSS(NSUB)=SUMARC
        NSUB=NSUB+1
        X=SOURC(NSUB)
C
C          IF THE SOURCE OF NSUB IS NOT A CUT NODE, WE NEED TO ADJUST NSUB
C          AND GO BACK TO THE MAIN PROGRAM FOR THE NEXT STAGE OF THE
C          BREAKUP
C
        IF (X.EQ.0) GO TO 31
        ORIGIN(1)=X
        GO TO 39
31      NSUB=NSUB-1
        SINKS(NSUB)=SINK
32      TNSUB=NCUT+1
        RETURN
        END
```

LOOP

This short program will determine whether a given network is acyclic or contains loops (cycles). The program examines each node and indicates whether or not the node is part of a loop. If a node is part of a loop, the number of activities in the loop is also indicated.

The basic steps in determining whether or not the INODE-th node is part of a loop are as follows:

(1) Identify all activities whose termanal node is the INODE-th node. Let A be the set of all origin nodes for these activities.

(2) If INODE is in A, the INODE-th node is part of a loop and stop.

(3) Identify all activities whose terminal node is in A. Redefine A to be the set of origin nodes for these activities. If A is now empty, stop and the INODE-th node is not part of a loop. If A is not empty, return to step 2.

Specific Input Instructions:

Card 1. Col. 1-3: The number of activities in the network, Format I3).
Col. 4-6: The largest node number in the network, Format (I3).

For each activity one card with:

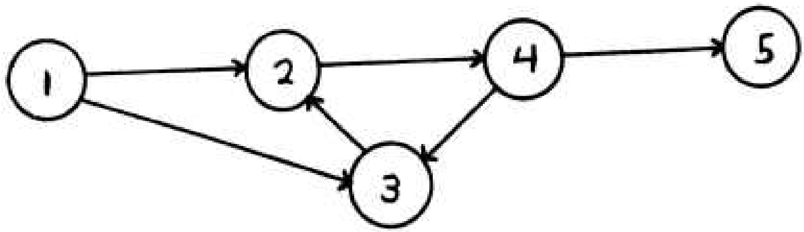
Col. 1-3: The activity's origin node number, Format (I3).
Col. 4-6: The activity's terminal node number, Format (I3).

The activities may be inputted in any order. The nodes may be numbered in any manner; however, the program is more efficient if the set of node numbers contains only the numbers 1 through N where N is the number of nodes in the network.

Dimension Restrictions:

This program is written in FORTRAN G. The current dimensions will allow a network with 300 activities and 200 nodes to be considered.

SAMPLE NETWORK



SAMPLE INPUT

```
6   5
1   2
1   3
4   3
3   2
2   4
4   5
```

SAMPLE OUTPUT

Activity	Tail	Head
1	1	2
2	1	3
3	4	3
4	3	2
5	2	4
6	4	5

```
Node     1 is not looped
Node     2 is looped.   There are 3 activities in the loop.
Node     3 is looped.   There are 3 activities in the loop.
Node     4 is looped.   There are 3 activities in the loop.
```

```
C          PROGRAM LOOP
C
C          THIS PROGRAM DETERMINES WHETHER THE INPUTTED NETWORK IS ACYCLIC
C          OR CONTAINS NODES WHICH ARE PART OF LOOPS (CYCLES)
C
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION A( 200),B( 200),HEAD(300),TAIL(300)
C
C          THE ARRAY DIMENSIONS ARE: A(N),B(N),HEAD(M),TAIL(M)
C          WHERE
C                    N = THE NUMBER OF NODES IN THE NETWORK
C                    M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C
      ILOOP=0
      READ (5,100) M,NMM
  100 FORMAT (2I3)
      READ(5,101) (TAIL(I),HEAD(I),I=1,M)
101   FORMAT(2I3)
      WRITE(6,2001)
 2001 FORMAT(1H1)
      WRITE(6,2000) (I,TAIL(I),HEAD(I),I=1,M)
 2000 FORMAT(' ACTIVITY     TAIL     HEAD'/,(4X,I3,7X,I3,6X,I3))
      WRITE(6,2001)
C     FORM THE 1ST HIERARCHY
      INODE = 0
   80 HIER = 2
      INODE = INODE + 1
      J = 0
      DO 1 I= 1,M
      IF(HEAD(I).NE.INODE) GO TO 1
      J = J+1
      A(J)=TAIL(I)
      IF (TAIL(I).EQ.INODE) GO TO 998
    1 CONTINUE
      IF (J.EQ.0) GO TO 997
      IA=J
      J=0
C     FORM THE SUBSEQUENT HIERARCHIES
  102 CONTINUE
      DO 2 II=1,IA
      DO 3 I=1,M
      IF (HEAD(I).NE.A(II)) GO TO 3
      IF(TAIL(I).EQ.INODE) GO TO 998
      IF(J.EQ.0) GO TO 40
      DO 10 K=1,J
      IF(TAIL(I).EQ.B(K)) GO TO 11
10    CONTINUE
40    CONTINUE
      J=J+1
      B(J)=TAIL(I)
11    CONTINUE
    3 CONTINUE
    2 CONTINUE
      IF (J.EQ.0) GO TO 997
      HIER = HIER+1
      IA=J
      J = 0
      DO 20 I=1,IA
20    A(I)=B(I)
```

```
      GO TO 102
  997 CONTINUE
      WRITE(6,2002) INODE
 2002 FORMAT(' NODE',I5,' IS NOT LOOPED' )
      IF (INODE.NE.NMM ) GO TO 80
      IF(ILOOP.EQ.1) GO TO 50
      WRITE(6,1000)
 1000 FORMAT (' THERE ARE NO LOOPS IN THIS NETWORK')
      GO TO 999
   50 WRITE(6,51)
51    FORMAT(' THERE ARE NO OTHER LOOPS IN THIS NETWORK')
      GO TO 999
  998 WRITE (6,1001) INODE,HIER
 1001 FORMAT (' NODE ' ,I5,' IS LOOPED. THERE ARE ',I3,' ACTIVITIES IN T
     *HE LOOP.')
      ILOOP=1
      INDDD=INODE+1
      IF(INDDD.NE.NMM) GO TO 80
999   CONTINUE
      WRITE(6,2001)
      STOP
      END
```

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Texas A&M University

2a. REPORT SECURITY CLASSIFICATION

unclassified

2b. GROUP

unclassified

3. REPORT TITLE

6 STATISTICAL PERT: DECOMPOSING A PROJECT NETWORK

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Progress Report

5. AUTHOR(S) (First name, middle initial, last name)

R. L. Sielken, Jr. [redacted] N. E. Fisher

10

12 44p.

6. REPORT DATE

11 Nov[redacted]75

7a. TOTAL NO. OF PAGES

37

7b. NO. OF REFS

0

8a. CONTRACT OR GRANT NO.

15 N00014-68-A-0140

b. PROJECT NO.

NR047-700

c.

16

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

Number 50

14 THEMIS-TR-50

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Office of Naval Research

13. ABSTRACT

See page following Table of Contents

9 THEMIS optimization research program,

347 380

ATTACHMENT III

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Statistical PERT						
Project Scheduling						
Network Decomposition						
ATTACHMENT III (Continued)						

BASIC DISTRIBUTION LIST FOR
UNCLASSIFIED TECHNICAL REPORTS

	Copies
Operations Research Office of Naval Research (Code 434) Arlington, Virginia 22217	3
Director, Naval Research Laboratory Attn: Library, Code 2029 (ONRL) Washington, D.C. 20390	2
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Defense Logistics Studies Information Exchange Army Logistics Management Center Fort Lee, Virginia 23801	1
Technical Information Center Naval Research Laboratory Washington, D.C. 20390	6
Office of Naval Research New York Area Office 207 West 24th Street New York, New York 10011 Attn: Dr. J. Laderman	1
Director, Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101 Attn: Dr. A.R. Laufer	1
Lt. Dennis Johnston - ED3 Manned Spacecraft Center NASA Houston, Texas 77058	1
Mr. William Dejka Code 360 Naval Electronics Laboratory Center San Diego, California 92132	1
Professor Lucien A. Schmit, Jr. Room 6731, Boelter Hall School of Engrng. & Applied Sci. University of California Los Angeles, California 90024	1
Director Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210 Attn: Dr. A.L. Powell	1
Director Office of Naval Research Branch Office 219 South Dearborn Street Chicago, Illinois 60604 Attn: Dr. A.R. Dawe	1
Office of Naval Research San Francisco Area Office 760 Market St. - Room 447 San Francisco, California 94103	1
Technical Library Naval Ordnance Station Indian Head, Maryland 20640	1
Bureau of Naval Personnel Navy Department Technical Library Washington, D.C. 20370 STOP 82	2
Library, Code 0212 Naval Postgraduate School Monterey, California 93940	1
Naval Ordnance Station Louisville, Kentucky 40214	1
Library Naval Electronics Laboratory Center San Diego, California 92152	1
Naval Ship Engineering Center Philadelphia Division Technical Library Philadelphia, Pennsylvania 19112	1
Dr. A.L. Slafkosky Scientific Advisor Commandant Marine Corps (Code AX) Washington, D.C. 20380	1

	Copies
Purdue University Graduate School of Ind. Admin. Lafayette, Indiana 47907 Attn: Prof. Andrew Whinston	1
Department of Economics Tisch Hall - 5th Floor Washington Square, NY Univ. New York City, NY 10003 Attn: Prof. O. Morgenstern	1
Dept. of Statistics Syntex Research 3401 Hillview Palo Alto, CA 94304 Attn: Prof. Stuart Bessler	1
University of California Department of Engineering Los Angeles, California 90024 Attn: Prof. R.R. O'Neill	1
Maritime Transportation Research Board National Academy of Sciences 2101 Constitution Avenue Washington, D.C. 20418 Attn: RADM J.B. Oren USCG (RET)	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. A.F. Veinott, Jr.	1
Texas A&M Foundation College Station, Texas 77843 Attn: Prof. H.O. Hartley	1
Case Western Reserve University Cleveland, Ohio 44106 Attn: Prof. B.V. Dean	1
University of California Center for Research in Management Science Berkeley, California 94720 Attn: Prof. W.I. Zangwill	1
U.S. Naval Postgraduate School Department of Operations Research and Economics Monterey, California 93940 Attn: Prof. C.R. Jones	

	Copies
Dr. Del Cilmer Naval Weapons Center (Code 607) China Lake, California 93555	1
Dr. Paul Murrill Project Themis Department of Chemical Engineering Louisiana State University Baton Rouge, Louisiana 70803	1
Director Army Materials & Mechanics Research Center Attn: Mr. J. Bluhm Watertown, Massachusetts 02172	1
Commanding Officer U.S. Army Ballistic Research Laboratories Attn: Dr. John H. Giese Aberdeen Proving Ground, Maryland 21005	1
University of Iowa Department of Mechanics and Hydraulics Iowa City, Iowa 52240 Attn: Prof. W.F. Ames	1
Office of Naval Research Resident Representative 1105 Guadalupe Lowich Building Austin, Texas 78701 Attn: Mr. Francis M. Lucas	1
Dr. Jerome Bracken Institute of Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1
Professor Richard L. Fox School of Engineering Case Western Reserve University Cleveland, Ohio 44106	1
Director, National Security Agency Ft. George G. Meade, Maryland 20755 Attn: Dr. Joseph Blum, R44	1

	Copies
Naval Applied Science Laboratory Technical Library, Code 222 Flushing & Washington Avenues Brooklyn, New York 11251	1
Naval Undersea Warfare Center 3202 E. Foothill Boulevard Pasadena, California 91107 Technical Library	1
University of Chicago Statistical Research Center Chicago, Illinois 60637 Attn: Prof. W. Kruskal	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. G.J. Lieberman	1
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. I.R. Savage	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. J.W. Tukey	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. T.W. Anderson	1
University of California Department of Statistics Berkeley, California 94720 Attn: Prof. P.J. Bickel	1
University of Washington Department of Mathematics Seattle, Washington 98105 Attn: Prof. Z.W. Birnbaum	1
Cornell University Department of Mathematics White Hall Ithaca, New York 14850 Attn: Prof. J. Kiefer	1

	Copies
Harvard University Department of Statistics Cambridge, Massachusetts 02139 Attn: Prof. W.G. Cochran	1
Columbia University Department of Industrial Engineering New York, New York 10027 Attn: Prof. C. Derman	1
New York University Institute of Mathematical Sciences New York, New York 10453 Attn: Prof. W.M. Hirsch	1
University of North Carolina Statistics Department Chapel Hill, North Carolina 27515 Attn: Prof. W.L. Smith and Prof. M.R. Leadbetter	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47079 Attn: Prof. H. Rubin	1
University of California, San Diego Department of Mathematics P.O. Box 109 La Jolla, California 92038 Attn: Prof. M. Rosenblatt	1
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. R.A. Bradley	1
New York University Department of Industrial Engineering & Operations Research Bronx, New York 10453 Attn: Prof. J.H.K. Kao	1
University of Wisconsin Department of Statistics Madison, Wisconsin 53706 Attn: Prof. G.E.P. Box	1

	Copies
Logistics Research Project The George Washington University 707 - 22nd Street, N.W. Washington, D. C. 20037 Attn: Dr. W. H. Marlow	1
International Business Machines Corporation P.O. Box 218, Lamb Estate Yorktown Heights, New York 10598 Attn: Dr. Alan Hoffman	1
University of California Management Sciences Research Project Los Angeles, California 90024 Attn: Dr. J.R. Jackson	1
Harvard University Department of Economics Cambridge, Massachusetts 02138 Attn: Prof. K.J. Arrow	1
Cowles Commission for Research in Economics Yale University New Haven, Connecticut 06520 Attn: Prof. Martin Shubik	1
Carnegie-Mellon University Graduate School of Industrial Administration Pittsburgh, Pennsylvania 15213 Attn: Prof. G. Thompson	1
University of California Department of Economics Berkeley, California 94720 Attn: Prof. R. Radner	1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. D. Gale	1
University of California Graduate School of Business Administration Los Angeles, California 90024 Attn: Prof. J. Marschak	1

	Copies
The University of Michigan Department of Mathematics, W.E. Ann Arbor, Michigan 48104 Attn: Prof. R.M. Thrall	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. A.W. Tucker	1
Case Western Reserve University Systems Research Center Cleveland, Ohio 44106 Attn: Prof. M. Mesarovic	1
University of Texas Department of Mathematics Austin, Texas 78712 Attn: Dr. A. Charnes	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Dr. D.L. Iglehart	1
University of Delaware Department of Mathematics Newark, Delaware 19711 Attn: Prof. R. Remage, Jr.	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. F.S. Hillier	1
Dr. Claude-Alain Burdet Asst. Prof. Industrial Admin. Carnegie-Mellon University Pittsburgh, Pennsylvania 15213	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. G. B. Dantzig	1
Chief of Naval Research (Code 436) Department of the Navy Arlington, Va. 22217	1
Science Librarian Kresge Library Oakland University Rochester, Michigan 48063	1

	Copies
University of Connecticut Department of Statistics Storrs, Connecticut 06268 Attn: Prof. H.O. Posten	1
Prof. Emanuel Parzen Statistics Department State Univ. New York at Buffalo 4230 Ridge Lea Road Amherst, New York 14226	1
ARCON Corporation Lakeside Office Park North Avenue at Route 128 Wakefield, Massachusetts 01880 Attn: Dr. A. Albert	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Chernoff	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. L.A. Savage	1
Rutgers-The State University Statistics Center New Brunswick, New Jersey 08903 Attn: Prof. H.F. Dodge	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. F. J. Anscombe	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47907 Attn: Prof. S.S. Gupta	1
Cornell University Department of Industrial Engineering Ithaca, New York 19850 Attn: Prof. R.E. Bechhofer	1
Mrs. Barbara Eaudi Univ. Program Coordinator, B.E. NASA Johnson Space Center Houston, TX 77058	1

	Copies
Professor Geoffrey S. Watson, Chairman Department of Statistics Princeton University Fine Hall Princeton, N. J. 08540	1
Stanford University Department of Mathematics Stanford, California 94305 Attn: Prof. S. Karlin	1
University of Sheffield Department of Probability and Statistics Sheffield 10, ENGLAND Attn: Prof. J. Gani	1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. R.E. Barlow	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Solomon	1
Applied Mathematics Laboratory Naval Ships Research Development Center Washington, D.C. 20007	1
Systems Analysis Division Room BE760, Pentagon Washington, D.C. 20350 Attn: Mr. A.S. Rhodes, Op-964	1
Department of Statistics University of North Carolina Chapel Hill, North Carolina 27515 Attn: Prof. M.R Leadbetter	1
Southern Methodist University Department of Statistics Dallas, Texas 75222 Attn: Prof. D.B. Owen	1
Israel Institute of Technology Technion Haifa, ISRAEL Attn: Prof. P. Naor	

Office of Gifts & Exchanges Library Texas A&M University College Station, Texas 77843	Copies 2
Archives Texas A&M University College Station, TX 77843	1